|  |  |
| --- | --- |
| Học trực tuyến CNTT, học lập trình từ cơ bản đến nâng cao |  |

**IOT304 – MẠNG TRUYỀN THÔNG KHÔNG DÂY**

**THUYẾT MINH ASSIGNMENT 2**

**KẾT HỢP VÀ SỬ DỤNG COMMAND LINE INTERFACE ĐỂ XỬ LÝ BÀI TOÁN BINDING, GROUPS TRONG ZIGBEE**

|  |  |
| --- | --- |
| Tên học viên:  Mã học viên:  Ngày báo cáo: | Dương Nhật Huy  FX29455  13/4/2024 |

HÀ NỘI, 2/2024

MỤC LỤC

[**1. Danh sách thư viện** 1](#_Toc164960536)

[**2. Danh sách các hàm** 1](#_Toc164960537)

[**3 . Hàm emberAfMainInitCallback(void)** 2](#_Toc164960538)

[**4. Hàm SEND\_BindingCommand()** 2](#_Toc164960539)

[**5. Hàm emberAfPreCommandReceivedCallback**() 2](#_Toc164960540)

[**6. Xử lý CLI** 2](#_Toc164960541)

NỘI DUNG THUYẾT MINH

# **1. Danh sách thư viện**

|  |  |  |
| --- | --- | --- |
| STT | Thư viện | Mô tả |
| 1 | af.h | Cung cấp các hàm và macro cơ bản cho các ứng dụng của Zigbee |
| 2 | led.h | Thư viện cung cấp các hàm xử lý Led |
| 3 | binding-table.h | Thư viện cung cấp các hàm tương tác với bảng binding |

# **2. Danh sách các hàm**

|  |  |  |
| --- | --- | --- |
| STT | Danh sách hàm | Mô tả |
| 1 | **void emberAfMainInitCallback(void)** | Hàm khởi động khi mới chạy chương trình |
| 2 | **void SEND\_BindingCommand(**  EmberAfClusterId ClusterID,  uint8\_t CommandId,  uint8\_t index,  uint8\_t source,  uint8\_t destination  ) | Gửi mệnh lệnh đến các thiết bị trong bảng Binding |
| 3 | **emberAfPreCommandReceivedCallback**(  EmberAfClusterCommand\* cmd) | Hàm ngắt mỗi khi nhận được mệnh lệnh từ thiết bị khác qua không khí và xử lý mệnh lệnh đó |

# **3 . Hàm emberAfMainInitCallback(void)**

*a. Mô tả*

Hàm được gọi mỗi khi khởi động chương trình. Gọi hàm ledInit() để cấu hình Led

# **4. Hàm SEND\_BindingCommand()**

*a. Mô tả*

Hàm gửi mệnh lệnh nhận được tới các phần tử trong bảng binding

*b. Tham số truyền vào*

EmberAfClusterId ClusterID là Cluster\_ID

uint8\_t CommandId là Command\_ID

uint8\_t index : là thứ tự trong bảng Binding

uint8\_t source: là endpoint nguồn

uint8\_t destination: là endpoint đích

# **5. Hàm emberAfPreCommandReceivedCallback**()

*a. Mô tả*

Hàm ngắt được gọi ra mỗi khi nhận lệnh. Có vai trò xử lý các lệnh nhận được

*b.Giá trị truyền vào*

EmberAfClusterCommand\* cmd con trỏ chỉ đến cấu trúc chứa các dữ liệu liên quan tới mệnh lệnh được gửi.

*c. Giá trị trả về*

Có kiểu dữ liệu boolean. Nếu giá trị trả về là False thì mệnh lệnh này sẽ tiếp tục được đưa xuống các tầng dưới để xử lý. Nếu giá trị là True thì kết thúc tại đây.

# **6. Xử lý CLI**

*a. Sơ đồ tổng quan của ZR*

A diagram of a flowchart

Description automatically generated

*b. Các bước thực xử lý CLI:*

*Bước 1:* Gia nhập mạng.

- Sử dụng lệnh CLI lên ZC plugin network-creator form [useCentralizedSecurity:1] [panId:2] [radioTxPower:1] [channel:1] để tạo mạng.

Trong đó:

+ useCentralizedSecurity nếu là 1 tức là tạo mạng centralized network, nếu là 0 thì tạo mạng Distributed network.

+ panId : Giá trị panID, định danh mạng.

+ radioTxPower: Transmission Power của mạng.

+ channel: kênh của mạng.

- Sử dụng plugin network-creator-security open-network trên ZC để mở mạng cho thiết bị khác gia nhập.

- Sử dụng plugin network-steering start [options:1] trên các thiết bị ZR để tiến hành gia nhập mạng.

Trong đó: options là một chuỗi 8 bit gồm các tùy chọn để chỉ ra hành vi cụ thể trong quy trình điều khiển mạng.

*Bước 2*: Điều khiển trạng thái On/Off LED của ZR từ ZC

- Sử dụng lệnh zcl on-off on để tạo một mệnh lệnh yêu cầu bật LED.

- Sử dụng lệnh zcl on-off off để tạo một mệnh lệnh yêu cầu tắt LED.

- Sử dụng lệnh send [id:2] [src-endpoint:1] [dst-endpoint:1] để gửi mệnh lệnh vừa tạo đến ZR

Trong đó:

+ id là shortID của ZR

+ src-endpoint là endpoint của ZC

+ dst-endpoint là endpoint của ZR

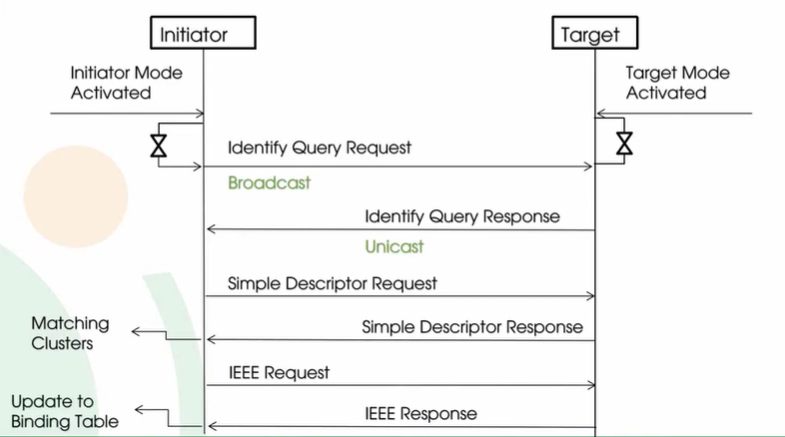
- Sau khi thiết bị nhận được bản tin, hàm **emberAfPreCommandReceivedCallback**() sẽ được gọi ra và xử mệnh lệnh nhận được

*Bước 3*: Tiến hành cấu hình Blinding hai thiết bị ZR1 & ZR2

- Sử dụng lệnh plugin find-and-bind initiator [endpoint:1] cho ZR1

- Sử dụng lệnh plugin find-and-bind target [endpoint:1] cho ZR2

Hai thiết bị sẽ tiến hành gửi các bản tin cho nhau để tiến hành cấu hình binding



- Sử dụng lệnh option binding-table print trên ZR1 xem bảng binding sau khi cấu hình xong.

A close-up of a computer code

Description automatically generated

- Trong đó:

+ type chính là số thứ tự trong bảng binding

+ nwk là phương thức giao tiếp

+ loc là endpoint của initatior

+ rem là endpoint của target

+ clus là cluster ID

+ node là short ID của thiết bị target

+ eui là long ID của thiết bị target

- Thực hiện bước hai lên lên ZC để gửi 1 bản tin on/off LED lên ZR1. Nếu cả hai ZR1 và ZR2 đều sáng thì đã thành công

*Bước 4:* Viết CLI để ZC sử dụng bản tin multicast gửi dữ liệu điều khiển đến các thiết bị trong Group

- Sử dụng zcl groups add [groupId:2] [groupName:-1] trên ZC để thêm thiết bị vào Group.

Trong đó

+ groupId là ID của Group

+ groupName là tên của Group

- Sử dụng send [id:2] [src-endpoint:1] [dst-endpoint:1] để gửi yêu cầu thêm thiết bị vào Group tới các thiết bị ZR.

- Sử dụng lệnh zcl on-off on và zcl on-off off để tạo mệnh lệnh on hoặc off led

- Sử dụng lệnh send\_multicast [groupId:2] [src-endpoint:1] trên ZC để gửi bản tin tới các thành viên trong Group.

Trong đó:

+ groupId là ID của Group

+ src-endpoint là endpoint của ZC